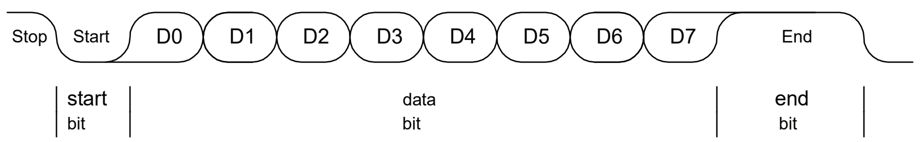
Communication Protocol

General MODBUS-RTU communication protocol for counting, timing and frequency measurement products

Ⅰ. MODBUS-RTU communication format

1. Basic rules
   1.1. Only one host is allowed in the same network.
   1.2. All RS485 communication channels should communicate in the master/slave mode.
   1.3. Any communication cannot be initiated by the slave.
   1.4. On the RS485 bus, all communications are transmitted in the form of "information frames". "Information frame" is a character string composed of a number of "data frames", which is a standard asynchronous serial data composed of an information header and sent coded data.
   1.5. If the host and slave receive an information frame containing an unknown command, they will not respond.
2. Transmission method
   Communication takes bytes (data frames) as the unit and uses asynchronous mode for transmission.
3. "Data frame" format
   Each "data frame" contains a start bit, 8 data bits, parity or no parity bit, a stop bit, a total of 10 bits of data.



4. "Information Frame" format

Address code	Function code	Data area	CRC check code
1 byte	1 byte	N byte	2 byte (Low byte first, high byte last)

When the communication command is sent from the master to the slave, the slave that matches the table address sent by the master receives the command. If the CRC check and the command format are correct, the slave executes the corresponding operation, and then returns the execution result to the master.

4.1. Address code (1 byte)

Contained in the address field of the "information frame", the address range is 1-247. The master strobes the slave by putting the address of the slave table into the address field of the command. When the slave machine returns data, it puts its own table address in the address field of the returned information. So that the master knows which slave responded (the table address of each device in the same bus must be unique).

4.1. Function code (1 byte)

Contained in the function code field of the "information frame". When sent from the master to the slave, the function code will inform the slave what operations need to be performed. When the slave responds, the function code is used to indicate the normal response or the occurrence of an error (abnormal response). For normal response, the slave will only return the received function code. For abnormal response, the slave changes the highest bit of the received function code to be 1 and then returns.

Function code definition

Function code	Definition	Operation
0x03	Read register	Read the data of a single or multiple registers
0x10	Write multiple registers	Write n data of 32-bit binary into n consecutive registers

4.3 Data area

Contained in the data field of the message, the length of the data varies depending on the function code.

4.4 CRC check code

Redundant cyclic code (CRC) contains 2 bytes, namely 16-bit binary. The CRC code is calculated by the sender and placed at the end of the sent message.The device at the receiving end recalculates the CRC code of the received information, and compares whether the calculated CRC code matches the received one. If the two do not match, it indicates an error.

The calculation method of CRC code is to preset all 16-bit registers first. Then gradually process every 8 bits of data information. When calculating the CRC code, only 8 data bits, the start bit and the stop bit are used. If there's parity check bit, it also includes the parity check bit, and they do not participate in the CRC code calculation.

When calculating the CRC code, the 8-bit data is XORed with the register data, and the result obtained is shifted by one bit to the lower, and the highest bit is filled with 0. Check the lowest bit again, if the lowest bit is 1, XOR the contents of the register with the preset number, if the lowest bit is 0, no XOR operation will be performed.

This process has been repeated 8 times. After the 8th shift, the next 8 bits are XORed with the contents of the current register. This process is repeated 8 times as above. When all the data information is processed, the final content of the register is the CRC code value.

CRC-16 code calculation steps

4.4.1. Set the 16-bit register to hexadecimal FFFF (that is, all 1s). Call this register the CRC register.
4.4.2. XOR an 8-bit data with the low bit of the 16-bit CRC register, and put the result in the CRC register.
4.4.3. Shift the contents of the register one bit to the right (toward the low bit), fill the highest bit with 0, and check the lowest bit (shift out bit).
4.4.4. If the lowest bit is 0: Repeat step 3 (shift again).
If the lowest bit is 1: The CRC register is XORed with the polynomial A001 (1010 0000 0000 0001).
4.4.5. Repeat steps 3 and 4 until the right shift is 8 times, so that the entire 8-bit data has been processed.
4.4.6. Repeat steps 2 to 5 to process the next 8-bit.
4.4.7. The finally obtained CRC register is the CRC code. (Low byte first, high byte last)

Ⅱ. Master command format and slave return message format

In order to support some hosts without 64-bit data types (such as some configuration software, PLC), the data in the 0x1000-0x105B address segment is uniformly enlarged by 2^{32} times. The purpose is to ensure the accuracy of the data and to make the integer part and the decimal part of the data can be handled separately.

2.1. read multiple registers

Example 1: Read count (time) value (complete data, 64-bit data format)

1. If the current count value of the meter = 123.456789, the host sends a command to read 4 registers starting from 0x1000, and the meter returns 0x7B74F01FB8.
2. Divide 0x7B74F01FB8 which is decimal 530242871224 by 2^{32} = the current count value of the slave is 123.456789

Command format	Host sends commands		Communication data word sequence =1234	=4321	=2143
Address area	Table address		0x01		
Function area	Function code		0x03		
Data area	Starting register address	High byte	0x10		
		Low byte	0x00		
	Number of read registers	High byte	0x00		
		Low byte	0x04		
Error detection area	CRC check code	Low byte	0x40		
		High byte	0xC9		

Message format	Slave return message		Communication data word sequence =1234	=2143	=4321
Address area	Table address		0x01		
Function area	Function code		0x03		
Data area	Number of data bytes		0x08		
	Count (timing) value register	High high byte	0x00	0x00	0x1F
			0x00	0x7B	0xB8
		High byte	0x00	0x00	0x74
			0x7B	0x00	0xF0
		Low byte	0x74	0x1F	0x00
			0xF0	0xB8	0x7B
		Low low byte	0x1F	0x74	0x00
			0xB8	0xF0	0x00
Error detection area	CRC check code	Low byte	0x62	0xFE	0xD6
		High byte	0x5C	0x65	0x28

Example 2: Read count (time) value (read-only integer part, 32-bit data format)

1. Assuming that the current count value of the slave = 19088743.568, read the integer part of the count value, and the slave returns data = 0x01234567.
2. When reading the integer part of a parameter individually, the return data 0x01234567=19088743 is the current actual value of the slave (no need to divide by 2^{32}).

Command format	Host sends commands		Communication data word sequence =1234	=2143	=4321
Address area	Table address		0x01		
Function area	Function code		0x03		
Data area	Starting register address	High byte	0x10	0x10	
		Low byte	0x00	0x02	
	Number of read registers	High byte	0x00	0x00	
		Low byte	0x02	0x02	
Error detection area	CRC check code	Low byte	0xC0	0x61	
		High byte	0xCB	0x0B	

Message format	Slave return message		Communication data word sequence =1234	=2143	=4321
Address area	Table address		0x01		
Function area	Function code		0x03		
Data area	Number of data bytes		0x04		
	Count (timing) value register	High byte	0x01	0x45	0x45
			0x23	0x67	0x67
		Low byte	0x45	0x01	0x01
			0x67	0x23	0x23
Error detection area	CRC check code	Low byte	0x79	0x1E	0x1E
		High byte	0x7F	0xA9	0xA9

2.2. Write multiple registers

Example 3: Write 12345.678 to the set value register of the slave PS2

1. If the host supports a 64-bit data format, you can directly multiply 12345.678 by 2^{32}=53024283256946, and then send it in hexadecimal format (53024283256946 = 0x00003039AD916872. There are 8 bytes in total, if not enough, fill in the high bits with 0)
2. If the host only supports the 32-bit data format, the integer part and the decimal part of 12345.678 need to be processed separately.
   2.1. The integer part does not need to be processed, directly put 12345 into the high 4 bytes of the data to be sent in hexadecimal format (if less than 4 bytes, fill in the high byte with 0. 12345 = 0x00003039).
   2.2. The decimal part 0.678 needs to be multiplied by 2^{32}= 2911987826, and put in the lower 4 bytes of the data to be sent in hexadecimal format (if less than 4 bytes, fill in the upper bit with 0. 2911987826 = 0x AD916872).
   2.3. Then send the processed 8 bytes of data in the order from high word to low word (1234) (0x00003039AD916872), or from low word to high word (4321) (0x6872AD9130390000).

Command format	Host sends commands		Communication data word sequence =1234	=2143	=4321
Address area	Table address		0x01		
Function area	Function code		0x10		
Data area	Starting register address	High byte	0x10		
		Low byte	0x30		
	Number of write registers	High byte	0x00		
		Low byte	0x04		
	Write data bytes		0x08		
	Ready to write the data in the PS2 setting value register (64-bit data, low byte first, high byte last)	High high byte	0x00	0x30	0x68
			0x00	0x39	0x72
		High byte	0x30	0x00	0xAD
			0x39	0x00	0x91
		Low byte	0xAD	0x68	0x30
			0x91	0x72	0x39
		Low low byte	0x68	0xAD	0x00
			0x72	0x91	0x00
Error detection area	CRC check code	Low byte	0x8F	0x63	0xA6
		High byte	0xFB	0xFA	0x4E

Message format	Slave return message		Communication data word sequence =1234	=2143	=4321
Address area	Table address		0x01		
Function area	Function code		0x10		
Data area	Starting register address	High byte	0x10		
		Low byte	0x30		
	Number of write registers	High byte	0x00		
		Low byte	0x04		
Error detection area	CRC check code	Low byte	0xC5		
		High byte	0x05		

Ⅲ. Communication error handling

When the meter detects errors other than the CRC check code error, it will return error information to the host. The slave machine sets the highest position of the received function code to 1, and then returns it as an error message together with the table address and error code.

3.1. The error code format that returned by the slave

Address code	Function code (highest position 1)	Error code	CRC check code low byte	CRC check code high byte
1 Byte	1 Byte	1 Byte	1 Byte	1 Byte

3.2. Error Code

0x01	Illegal function code	The received function code is not supported by the instrument
0x02	Illegal register address	The received register address exceeds the instrument register address range
0x03	Illegal number of registers	The number of registers received exceeds the number of instrument registers
0x04	Illegal data value	The received data value exceeds the data range of the corresponding address

Ⅳ. Data and mapping address

4.1. The data of each parameter in the 0x1000-0x105B address segment is uniformly enlarged by 2^{32} times, it needs to be multiplied by 2^{32} before writing, and divided by 2^{32} when reading.

4.2. Each parameter in the 0x1000-0x105B address segment occupies 4 register addresses (4 words, 8 bytes), the internal data is divided into 1234(default), 4321 and 2143 in the unit of register.

4.3. This protocol is a general communication protocol. Please refer to the corresponding product operation manual for whether the instrument has the functions in the protocol and the value range of the register.

* **Note:** The data address of the meter is usually expressed in hexadecimal. But when the PLC is used as the host to communicate with the meter, the data address (using the "register number" in the PLC application, which is the decimal value corresponding to the hexadecimal number) needs to be converted into the PLC format address by adding 40001. Taking Siemens S7-200 series PLC as an example, when operating the meter 0x1000 (register number 4096) address data, the corresponding address of the PLC is equal to "4096" (corresponding to 0x1000) + 40001 = 44097.

No.	Add（Register number*）	Parameter	Length	Data type	Attributes	Remark
1	0x1000（44097） 0x1001（44098） 0x1002（44099） 0x1003（44100）	Counting (timing) value	4	Signed 64-bit integer	R/W	1. During write operation, only 0 can be written, otherwise an error will be returned. 2. In timing mode, the unit is second. Example: Register value = 0xCE3D70A3D Actual time = 0xCE3D70A3D / 2^{32} = 12.89 seconds
2	0x1004（44101） 0x1005（44102） 0x1006（44103） 0x1007（44104）	Batch or total value	4	Signed 64-bit integer	R/W	1. When writing, only 0 can be written, otherwise an error will be returned
3	0x1008（44105） 0x1009（44106） 0x100a（44107） 0x100b（44108）	Frequency, rpm, linear speed value	4	Signed 64-bit integer	R	
4	Reserved					
5	0x1010（44113） 0x1011（44114） 0x1012（44115） 0x1013（44116）	Initial value of the counting	4	Signed 64-bit integer	R/W	
6	0x1014（44117） 0x1015（44118） 0x1016（44119） 0x1017（44120）	Counting coefficient value	4	Signed 64-bit integer	R/W	
7	0x1018（44121） 0x1019（44122） 0x101a（44123） 0x101b（44124）	Line speed or batch factor value	4	Signed 64-bit integer	R/W	
8	Reserved					
9	0x1020（44129） 0x1021（44130） 0x1022（44131） 0x1023（44132）	PS1 setting value	4	Signed 64-bit integer	R/W	
10	0x1024（44133） 0x1025（44134） 0x1026（44135） 0x1027（44136）	PS1 output delay	4	Signed 64-bit integer	R/W	1. Unit: second 2. Used as output close delay register in some frequency meters.
11	0x1028（44137） 0x1029（44138） 0x102a（44139） 0x102b（44140）	PS1 backlash	4	Signed 64-bit integer	R/W	Only some frequency meters have this parameter
12	0x102c（44141） 0x102d（44142） 0x102e（44143） 0x102f（44144）	PS1 output start delay	4	Signed 64-bit integer	R/W	1. Unit: second 2. Only some frequency meters have this parameter
13	0x1030（44145） 0x1031（44146） 0x1032（44147） 0x1033（44148）	PS2 setting value	4	Signed 64-bit integer	R/W	In timing mode: the unit is second, and the setting range is determined by the current meter timing range. If the meter's timing range=99H59M59S99, the writable range of the register=0.01~359999.99S
14	0x1034（44149） 0x1035（44150） 0x1036（44151） 0x1037（44152）	PS2 output delay	4	Signed 64-bit integer	R/W	1. Unit: second 2. As output in some frequency meters Turn off the delay register.
15	0x1038（44153） 0x1039（44154） 0x103a（44155） 0x103b（44156）	PS2 backlash	4	Signed 64-bit integer	R/W	Only some frequency meters have this parameter
16	0x103c（44157） 0x103d（44158） 0x103e（44159） 0x103f（44160）	PS2 output start delay	4	Signed 64-bit integer	R/W	1. Unit: second 2. Only some frequency meters have this parameter.
17	0x1040（44161） 0x1041（44162） 0x1042（44163） 0x1043（44164）	LSV setting value	4	Signed 64-bit integer	R/W	
18	0x1044（44165） 0x1045（44166） 0x1046（44167） 0x1047（44168）	LSV output delay time	4	Signed 64-bit integer	R/W	Unit: second
19	0x1048（44169） 0x1049（44170） 0x104a（44171） 0x104b（44172）	LSV backlash	4	Signed 64-bit integer	R/W	
20	Reserved					
21	0x1050（44177） 0x1051（44178） 0x1052（44179） 0x1053（44180）	BAS setting value	4	Signed 64-bit integer	R/W	
22	0x1054（44181） 0x1055（44182） 0x1056（44183） 0x1057（44184）	BAS output delay time	4	Signed 64-bit integer	R/W	Unit: second
23	0x1058（44185） 0x1059（44186） 0x105a（44187） 0x105b（44188）	BAS backlash	4	Signed 64-bit integer	R/W	

4.4. Each parameter in the 0x1100-0x1164 address segment occupies 1 register address (1 word, 2 bytes), data in the register is high byte first low byte last.

No.	Add (register number)	Parameter	Length	Data type	Attributes	Remark
24	0x1100（44353）	Communication address	1	Unsigned 16-bit integer	R/W	1~247
25	0x1101（44354）	Reserved				
26	0x1102（44355）	Reserved				
27	0x1103（44356）	Communication baud rate	1	Unsigned 16-bit integer	R/W	4800=4800bit/s、9600=9600bit/s、19200=19200bit/s
28	0x1104（44357）	Communication verification method	1	Unsigned 16-bit integer	R/W	0=no parity, 1=odd parity, 2=even parity
29	0x1105（44358）	Communication data (register) Order selection	1	Unsigned 16-bit integer	R/W	Example: When sending or receiving data 0x1020304050607080, the corresponding sequence of different settings is as follows: =1234, the order of receiving and sending = 10 20 30 40 50 60 70 80 =2143, the order of receiving and sending = 30 40 10 20 70 80 50 60 =4321, the order of receiving and sending = 70 80 50 60 30 40 10 20
30	0x1106（44359）	Batch/total accumulation method selection	1	Unsigned 16-bit integer	R/W	0=accumulate by batch, 1=accumulate by total
31	0x1107（44360）	Function selection	1	Unsigned 16-bit integer	R/W	0=counting, 1=timing, 2=frequency, 3=rpm, 4=line speed
32	0x1108（44361）	Ascending or descending method selection	1	Unsigned 16-bit integer	R/W	0=rising, 1=falling
33	0x1109（44362）	NPN,PNP selection	1	Unsigned 16-bit integer	R/W	0=NPN，1=PNP
34	0x110a（44363）	Input type selection	1	Unsigned 16-bit integer	R/W	0=U, 1=D, 2=UD-A, 3=UD-B, 4=UD-C, 5=UD-D
35	0x110b（44364）	Input frequency selection	1	Unsigned 16-bit integer	R/W	1=1Hz，30=30Hz，1000=1KHz，5000=5 KHz，10000=10KHz 20000=20KHz
36	0x110c（44365）	External signal width selection	1	Unsigned 16-bit integer	R/W	Actual pulse width, unit: ms
37	0x110d（44366）	Reserved				
38	0x110e（44367）	Reserved				
39	0x110f（44368）	Timing range selection	1	Unsigned 16-bit integer	R/W	0 = 999999s99 256 = 99h59m59s99 512 = 9999h59m59s Remarks: In non-timer relay or timing mode, writing is invalid
40	0x1110（44369）	Delay range selection	1	Unsigned 16-bit integer	R/W	
41	0x1111（44370）	Reserved				
42	0x1112（44371）	Display decimal point selection	1	Unsigned 16-bit integer	R/W	0=no decimal point or floating decimal point, 1=1 decimal point, 2=2 decimal point,...
43	0x1113（44372）	Display refresh time selection	1	Unsigned 16-bit integer	R/W	Unit (10ms): 0=auto refresh, 50=0.5 second, 100=1 second
44	0x1114（44373）	Reserved				
45	0x1115（44374）	Reserved				
46	0x1116（44375）	Counting output mode selection	1	Unsigned 16-bit integer	R/W	0 = F, 1 = N, 2 = C, 3 = R 4 = K, 5 = P, 6 = Q, 7 = A 8 = S, 9 = T, 10 = D, 11 = M
47	0x1117（44376）	Timing output mode selection	1	Unsigned 16-bit integer	R/W	0 = OND, 1 = OND.1, 2 = OND.2, 3 = FLK 4 = FLK.1, 5 = FLK.2, 6 = INT, 7 = INT.1 8 = OFD
48	0x1118（44377）	SV1 output mode selection (reserved)	1	Unsigned 16-bit integer	R/W	
49	0x1119（44378）	SV2 output mode selection (reserved)	1	Unsigned 16-bit integer	R/W	
50	0x111a（44379）	SV3 output mode selection (reserved)	1	Unsigned 16-bit integer	R/W	
51	0x111b（44380）	LSV output mode selection (reserved)	1	Unsigned 16-bit integer	R/W	
52	0x111c（44381）	BSV output mode selection (reserved)	1	Unsigned 16-bit integer	R/W	
53	0x111d（44382）	Power failure memory function	1	Unsigned 16-bit integer	R/W	0=off, 1=on
54	0x111e（44383）	Start function	1	Unsigned 16-bit integer	R/W	0=off, 1=on
55	0x111f（44384）	Reserved				
56	0x1120（44385）	Reserved				
57	0x1121（44386）	Reserved				
58	0x1122（44387）	Password setting	1			
59	0x1160（44449）	OUT1 output status	1	Unsigned 16-bit integer	R	0=no action, 1=action
60	0x1161（44450）	OUT2 output status	1	Unsigned 16-bit integer	R	0=no action, 1=action
61	0x1162（44451）	OUT3 output status	1	Unsigned 16-bit integer	R	0=no action, 1=action
62	0x1163（44452）	LSO output status	1	Unsigned 16-bit integer	R	0=no action, 1=action
63	0x1164（44453）	BAO output status	1	Unsigned 16-bit integer	R	0=no action, 1=action